

RELAÇÃO ENTRE MODELOS DE PROGRAMAÇÃO NÃO-LINEAR COM INCERTEZA NO CONJUNTO DE RESTRIÇÕES

Ricardo Coêlho Silva*

Departamento de Telemática / FEEC
Universidade Estadual de Campinas (UNICAMP)
Campinas – SP
rcoelhos@dt.fee.unicamp.br; rcoelhos@gmail.com

Luiza Amalia Pinto Cantão

Departamento de Engenharia Ambiental
Universidade Estadual Paulista (UNESP)
Sorocaba – SP
luiza@sorocaba.unesp.br

Akebo Yamakami

Departamento de Telemática / FEEC
Universidade Estadual de Campinas (UNICAMP)
Campinas – SP
akebo@dt.fee.unicamp.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 06/2006; aceito em 10/2008 após 3 revisões
Received June 2006; accepted October 2008 after 3 revisions

Resumo

Este trabalho demonstra de forma analítica e numérica a relação entre dois métodos, Trappey *et al.* (1988) e Xu (1989), da literatura que resolvem problemas de programação não-linear com incertezas no conjunto de restrições. Uma análise comparativa entre o desempenho para a obtenção da solução ótima dos métodos de otimização não-linear clássicos e dos métodos de otimização não-linear nebulosos, apresentados também neste trabalho. Para tal comparação, serão apresentados dois problemas que foram modelados em termos de programação não-linear clássico, os quais permitem a introdução de incertezas nas restrições. Com base na análise dos problemas propostos em Xu (1989), verificou-se que os dois métodos descritos fornecem resultados similares, conforme algumas condições.

Palavras-chave: conjuntos nebulosos; otimização não-linear; programação matemática nebulosa.

Abstract

In this work, we demonstrate analytical and numerically the relation between two methods, Trappey *et al.* (1988) and Xu (1989), found in the literature. These methods were developed to solve nonlinear programming problems with uncertainties in the set of constraints. A comparative analysis of the performance between classic and fuzzy nonlinear optimization methods are presented too in the work. For the comparison, two problems are modeled that had been shaped in terms of classic nonlinear programming, which allow the introduction of uncertainties in its formularizations. Based on the analysis of the problem proposed in Xu (1989), we verified that the two described methods provide similar results, as per some conditions.

Keywords: fuzzy set; nonlinear optimization; fuzzy mathematical programming.

1. Introdução

Pesquisa Operacional (PO) é uma área da ciência que desenvolve técnicas para otimizar a realização de sistemas. O aparecimento da PO forneceu uma base quantitativa e racional para uma tomada de decisão. O estudo de PO pode ser dividido em várias subáreas e programação matemática é uma delas. Em geral, os problemas de programação matemática (PPM) envolvem minimização (ou maximização) da função objetivo sujeito a um conjunto de restrições. Quando definimos problemas de programação matemática clássica (PPMC), as suas modelagens e as decisões são mais precisas. Um problema convencional é modelado na forma de PPMC se o problema tem definições matemáticas breves e claras, da seguinte forma:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.a} \quad & g_k(x) \leq b_k, \quad k = 1, 2, \dots, n \\ & h_l(x) = c_l, \quad l = 1, 2, \dots, m \\ & x \geq 0 \end{aligned} \tag{1}$$

Os modelos de otimização usam programação matemática tradicional, os quais tentam representar as operações importantes construindo modelos matemáticos exatos. Contudo, não conseguem representar toda a realidade presente. A Lógica Nebulosa (Zadeh, 1965) tem mostrado um grande potencial para a modelagem desses sistemas, os quais podem ser não-lineares, complexos e mal-definidos. A Lógica Nebulosa estabeleceu numerosas aplicações devido a seu alívio de implementação, flexibilidade, tolerância natural para dados imprecisos, e habilidade para modelar modelos não-lineares de complexidade arbitrária.

A programação matemática nebulosa pode ter incertezas na relação entre os números difusos, presentes no conjunto de restrições e/ou na função objetivo, veja Lee *et al.* (1999), Trappey *et al.* (1988) e Xu (1989), e nos parâmetros do problema, sendo que esses parâmetros podem representar as constantes e/ou as variáveis de decisão, veja Ekel (2002) e Ekel (2005).

Em adição às características descritas anteriormente, um problema de programação nebuloso pode também ser dividido em programação flexível, que resolve os problemas nebulosos usando a teoria de representação inicialmente publicada por Bellman & Zadeh em 1970, e programação possibilística ou robusta, descrito em Lai & Hwang (1992), que associa os dados imprecisos a uma função de distribuição de possibilidade que pode ser definida pela teoria de possibilidade. As distribuições de possibilidade são análogas às distribuições de probabilidades e podem ser subjetivos ou objetivos. Neste trabalho focamos as pesquisas na programação flexível com a incerteza presente na relação entre números nebulosos no conjunto de restrições. Contudo, pouco destes estudos se intensificaram na área de programação não-linear com incertezas no conjunto de restrições em espaço contínuo, dentre eles podemos referenciar Silva & Yamakami (2004) e Silva & Cantão (2004).

Um conjunto nebuloso é definido por uma função de pertinência $\mu_A(x)$, que estabelece para cada x um grau de pertinência ao conjunto A , com $\mu_A \in [0,1]$. Assim, podemos expressar matematicamente o conjunto nebuloso como:

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x - a_{\inf}}{a_{\text{mod}} - a_{\inf}} & x \in [a_{\inf}, a_{\text{mod}}] \\ \frac{a_{\text{sup}} - x}{a_{\text{sup}} - a_{\text{mod}}} & x \in [a_{\text{mod}}, a_{\text{sup}}] \\ 0 & \text{caso contrário} \end{cases}$$

sendo a_{mod} o valor modal, a_{\inf} e a_{sup} são os limitantes inferior e superior, respectivamente. A Figura 1 ilustra a função de pertinência descrita pela equação acima.

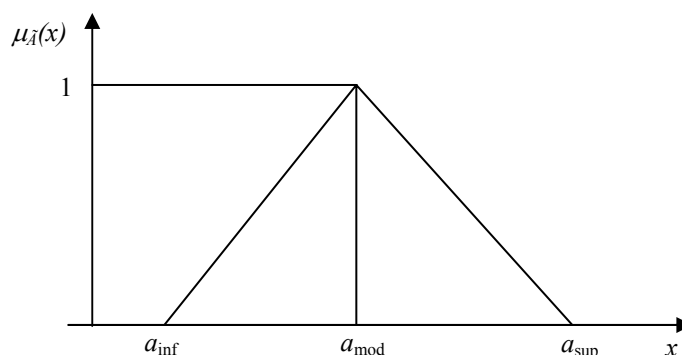


Figura 1 – Função de pertinência $\mu_{\tilde{A}}(x)$.

Este trabalho está organizado em 7 seções. A próxima Seção estão apresentados os tipos de restrições usadas em problemas de programação matemática, sendo encontradas representações geométricas da introdução de incertezas nessas restrições de otimização. Na Seção 3 são apresentados 2 métodos que abordam algumas definições básicas de programação matemática nebulosa para resolver problemas de programação não-linear nebulosos (PPNLN), que transforma os PPNLN em problemas de programação não-linear clássicos (PPNLC). O primeiro transforma a função objetivo em uma restrição e a nova função objetivo irá maximizar o nível de satisfação dentro do espaço de factibilidade. O segundo é dividido em duas fases: Na primeira fase, usa-se o método de níveis de α -cortes transformando o PPNLN em um PPNLC, para determinar um conjunto domínio para o problema. Na segunda fase, é aplicado o método de busca nos limites dos conjuntos de soluções factíveis da fase 1, sendo que utilizando o supremo e o ínfimo da solução da fase anterior, obtém-se um intervalo de confiança que contém a solução nebulosa desejada no espaço de discurso. Na Seção 4 é apresentada uma relação entre os métodos descritos na Seção 3. A Seção 5 descreve alguns exemplos numéricos utilizados neste artigo para fins de comparação entre os métodos clássicos e nebulosos. A Seção 6 apresenta os resultados computacionais dos exemplos numéricos da seção anterior. Na última Seção realiza-se uma análise do conjunto de resultados computacionais obtidos.

2. Conjunto de restrições nebulosas

Existem três tipos de problemas de programação matemática com restrições: (a) com conjunto de restrições de igualdade; (b) com conjunto de restrições de desigualdade e (c) com conjunto de restrições mistas, isto é, com restrições de igualdade e desigualdade no mesmo problema.

As restrições de igualdade clássicas podem ser representadas graficamente da seguinte forma:

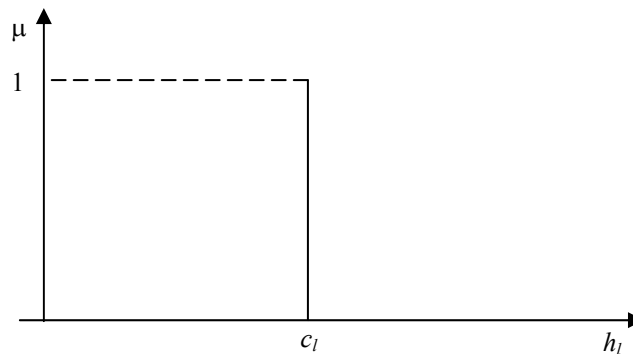


Figura 2 – Representação geométrica da restrição de igualdade.

Enquanto as restrições de desigualdade clássicas podem ser representadas graficamente da seguinte forma:

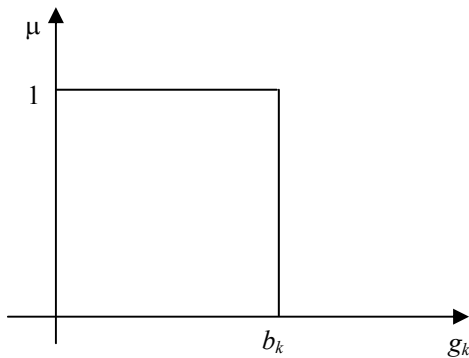


Figura 3 – Representação geométrica da restrição de desigualdade decrescente.

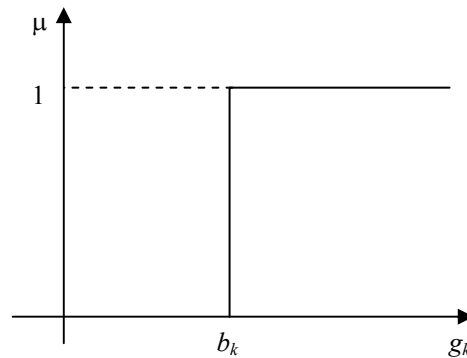


Figura 4 – Representação geométrica da restrição de desigualdade crescente.

A construção de um problema de programação matemática com restrições nebulosas é baseada no modelo geral que pode ser expresso da seguinte forma:

$$\begin{aligned}
 & \tilde{\min} \quad f(x) \\
 & \text{s.a} \quad g_i(x) \lesssim b_i, \quad i = 1, 2, \dots, m \\
 & \quad \quad h_j(x) \cong c_j, \quad j = 1, \dots, n \\
 & \quad \quad x \geq 0
 \end{aligned} \tag{2}$$

sendo que o símbolo “ \sim ” indica que as restrições contêm informações incertas, e podem sofrer um certa violação, que representa uma variação no termo independente permitindo um relaxamento nas restrições do problema. Nos problemas abordados neste trabalho, a natureza nebulosa está inserida nos símbolos, ou seja, nos sinais de igualdade e de desigualdade de cada restrição.

No Problema (2) pode-se permitir tolerâncias nas restrições, isto é, para os valores admissíveis b_i e b_j são dados, para cada restrição, violações máximas toleradas. Logo, este problema pode ser reescrito da seguinte forma:

$$\begin{aligned} & \sim \\ \min & \quad g_0(x) \\ \text{s.a} & \quad g_i(x) \leq b_i + T_i, \quad i = 1, 2, \dots, m+n \\ & \quad g_j(x) \geq b_j - T_j, \quad j = 1, \dots, n \\ & \quad x \geq 0 \end{aligned} \tag{3}$$

Em outras palavras, cada restrição nebulosa corresponde a subconjunto nebuloso no universo de números reais, definido pela função de pertinência $\mu(x): R^n \rightarrow [0, 1]$.

As funções de pertinência associadas às restrições de igualdade nebulosas são apresentadas da seguinte forma:

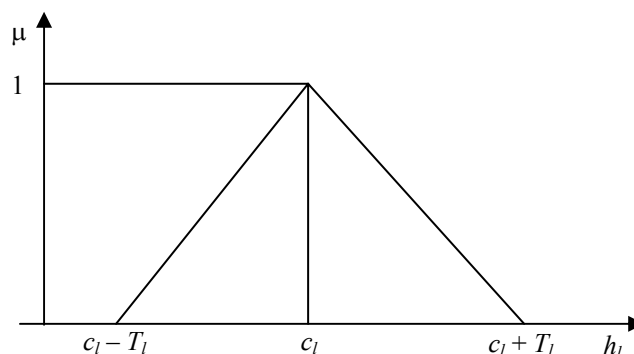


Figura 5 – Função de pertinência da restrição de igualdade.

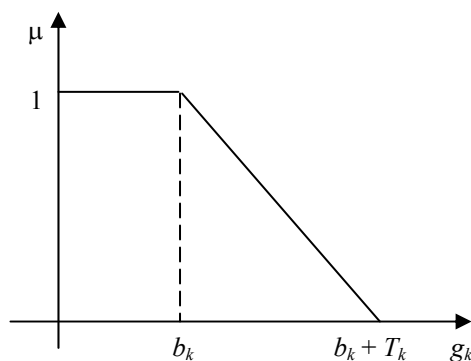


Figura 6 – Função de pertinência da restrição de desigualdade decrescente.

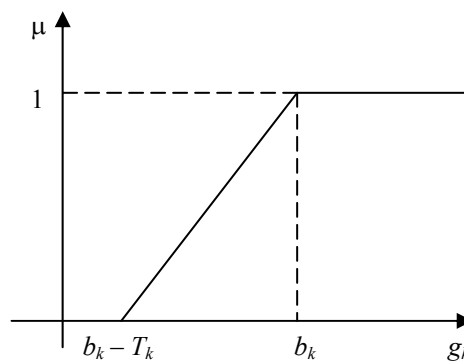


Figura 7 – Função de pertinência da restrição de desigualdade crescente.

Já as funções de pertinências associadas às restrições de desigualdade nebulosas podem ser escritas de duas formas que estão representadas graficamente nas Figuras 6 e 7.

As Figuras 5, 6, 7 mostram geometricamente a transformação de um problema de programação matemático nebuloso para um problema de programação matemático clássico. Essas transformações podem ser escritas de forma analítica, denominada função de pertinência da seguinte forma:

(a) Função de pertinência da restrição de igualdade

$$\mu_i(x) = \begin{cases} \frac{h_i(x) - (c_i + T_i)}{T_i}, & \text{se } c_i - T_i < h_i(x) \leq c_i \\ \frac{(c_i + T_i) - h_i(x)}{T_i}, & \text{se } c_i < h_i(x) < c_i + T_i \\ 0, & \text{caso contrário} \end{cases}$$

(b) Função de pertinência da restrição de desigualdade decrescente

$$\mu_i(x) = \begin{cases} 0, & \text{se } g_i(x) \geq b_i + T_i \\ \frac{(b_i + T_i) - g_i(x)}{T_i}, & \text{se } b_i < g_i(x) < b_i + T_i \\ 1, & \text{se } g_i(x) \leq b_i \end{cases}$$

(c) Função de pertinência da restrição de desigualdade crescente

$$\mu_i(x) = \begin{cases} 0, & \text{se } g_i(x) \leq b_i - T_i \\ \frac{g_i(x) - (b_i - T_i)}{T_i}, & \text{se } b_i - T_i < g_i(x) < b_i \\ 1, & \text{se } g_i(x) \geq b_i \end{cases}$$

Os métodos descritos neste trabalho solucionam somente problemas com restrições de desigualdade. Assim, cada restrição de igualdade é reescrita em duas restrições de desigualdade. Portanto, a Figura 5 é dividida nas Figuras 6 e 7, com $c_i = b_k$ e $T_i = T_k$. Devido à decomposição das restrições de igualdade, o número de restrições do problema (2) passa a ter $m + 2n$ restrições, sendo m o número de restrições de desigualdade do problema original e $2n$ representa as restrições de desigualdades geradas pelas n restrições de igualdade.

A obtenção de mais informações sobre os métodos de programação matemática nebulosa, que serão utilizados neste artigo podem ser encontrados em Silva & Yamakami (2004), Silva & Cantão (2004) e Silva (2005), e as idéias principais para a formulação dos dados de incerteza para os exemplos numéricos apresentados na Seção 3, podem ser encontrados em Trappey *et al.* (1988) e Xu (1989).

3. Programação não-linear nebulosa

Nesta Seção estão apresentados métodos iterativos desenvolvidos para solucionar problemas de programação não-linear nebulosos. As características nebulosas podem aparecer de várias formas no problema, podendo o mesmo ser totalmente nebuloso. Neste trabalho foi introduzida incerteza no conjunto de restrições. Na Subseção 3.1 será mostrado o primeiro método de programação não-linear nebuloso, encontrado em Lee *et al.* (1999) e Trappey *et al.* (1988), que originou-se do método de programação linear nebuloso idealizado por Zimmermann (1983). Na Subseção 3.2 é mostrado o segundo método, desenvolvido por Xu (1989), o qual utiliza duas fases para obter a solução ótima.

3.1 Primeiro método

Trataremos aqui do método que transforma um problema de programação não-linear nebuloso (PPNLN) em um problema de programação não-linear clássico (PPNLC). Esse método transforma a função objetivo em uma restrição, e a nova função objetivo tenta maximizar o nível de compatibilidade no espaço de factibilidade. Este nível de compatibilidade está determinado no intervalo real $[0,1]$, e diante de seu valor saberemos o quanto foram satisfeitas as restrições.

Um problema de programação não-linear com uma função objetivo nebulosa e restrições nebulosas pode ser escrito genericamente, utilizando como base o problema (4), da seguinte forma

$$\begin{aligned} & \tilde{\min} \quad g_0(x) \\ & \text{s.a} \quad g_i(x) \tilde{<} b_i, \quad i = 1, 2, \dots, m, m+1, \dots, n \\ & \quad \quad x \geq 0 \end{aligned} \quad (4)$$

sendo que ‘ $\tilde{\min}$ ’ e ‘ $\tilde{<}$ ’ representam as características nebulosas do problema. Pode-se notar que a formulação sem ‘ $\tilde{}$ ’ é o modelo de programação não-linear clássico original. A função objetivo com o sinal nebuloso pode ser interpretada como

$$\begin{aligned} & g_0(x) \tilde{<} b_0 \\ & g_i(x) \tilde{<} b_i \quad i = 1, 2, \dots, m, m+1, \dots, n \\ & \quad \quad x \geq 0 \end{aligned}$$

sendo que g_0 deve ser no mínimo menor que b_0 . Desta forma, acrescenta-se uma variável t_i que indica o nível de violação das restrições nebulosas, então pode-se modificar o problema (4) da seguinte forma:

$$\begin{aligned} & g_j(x) < b_j + t_j \quad j = 0, 1, 2, \dots, m, m+1, \dots, n \\ & \quad \quad x \geq 0 \\ & \quad \quad 0 \leq t_j \leq T_j \end{aligned}$$

sendo que $t_j = [t_0, t_1, \dots, t_n]$ representa a variação de violação da restrição j , e $T_j = [T_0, T_1, \dots, T_n]$ representa o valor máximo de violação admissível da restrição j .

A decisão tem de satisfazer a função de pertinência $\mu_j(g_j(x))$ descrita abaixo

$$\mu_j(g_j(x)) = \begin{cases} 0, & \text{se } t_j = T_j \\ 1 - \frac{t_j}{T_j}, & \text{se } 0 < t_j < T_j \\ 1, & \text{se } t_j = 0 \end{cases}$$

sendo que cada função de pertinência representa o grau de satisfação que a solução x fornece para a restrição j . Contudo, pode-se definir uma intersecção de todas as restrições como

$$\bigcap_{0 \leq j \leq n} \bigcap_x \mu_j(g_j(x)) \in [0,1] \\ 0 \leq t_j \leq T_j$$

sendo \bigcap o símbolo de intersecção.

Usando a regra de minimização para a intersecção de múltiplos conjuntos nebulosos, a primeira decisão feita tenta achar a intersecção satisfatória das pertinências dentro de cada restrição. Então, a decisão feita tenta maximizar o grau da união de todas as restrições satisfeitas. Fazendo $S = \min_{0 \leq j \leq n} \mu_j(g_j(x))$, o modelo não-linear pode ser formulado como um

PPNL Clássico padrão. Assim,

$$\begin{aligned} \max \quad & S \\ \text{s.a.} \quad & (a) \quad T_j S + t_j \leq T_j \\ & (b) \quad g_j(x) - t_j \leq b_j \\ & (c) \quad t_j \leq T_j \\ & S, x, t_j \geq 0 \quad j = 0, 1, 2, \dots, m, m+1, \dots, n \end{aligned} \quad (5)$$

Note que a função objetivo S e as restrições (a) e (c) são lineares. A restrição (b) é não-linear.

3.2 Método duas fases

O Método Duas Fases começa transformando um PPNL Nebuloso em um PPNL Clássico equivalente. Então, podemos escrever o problema (4) como

$$\begin{aligned} \min \quad & g_0(x) \\ \text{s.a.} \quad & g_i(x) \leq b_i + t_i, \quad i = 1, 2, \dots, m, m+1, \dots, n \\ & x \geq 0 \end{aligned} \quad (6)$$

Contudo, utilizando uma função $\mu_j(x): IR^n \rightarrow [0,1]$, teremos graus diferentes de satisfação dentro do intervalo unitário $[0, 1]$, da seguinte forma

$$\mu_i(x) = \begin{cases} 0, & \text{se } g_i(x) \geq b_i + T_i \\ \frac{(b_i + T_i) - g_i(x)}{T_i}, & \text{se } b_i < g_i(x) < b_i + T_i \\ 1, & \text{se } g_i(x) \leq b_i \end{cases}$$

Para tanto, precisa-se parametrizar as restrições usando os níveis α -cortes propostos na Teoria de Conjuntos Nebulosos (Zadeh, 1965), como

$$C_\alpha = \{x \mid x \in IR^n, \mu_C(x) \geq \alpha\}, \forall \alpha \in [0,1].$$

Depois de feito a transformação e os cortes, realiza-se a intersecção de todas as restrições, com um operador de agregação, da seguinte forma

$$\mu_C(x) = \inf_{i=1}^n \mu_i(x), \forall x \in IR^n.$$

Logo, utiliza-se o ínfimo como operador de agregação, pois não se pode garantir que o conjunto é finito. Por fim, consegue-se escrever o PPNLN em um PPNLC correspondente. Contudo, existem vários valores da nossa função objetivo, tantos quantos forem o tamanho da parametrização de α . Logo, o Problema (6) é reescrito como

$$\begin{aligned} & \sim \\ \min & g_0(x) \\ \text{s.a.} & g_i(x) \leq b_i + T_i(1-\alpha), \quad i = 1, 2, \dots, m, m+1, \dots, n \\ & x \geq 0 \quad \alpha \in [0,1] \end{aligned} \quad (7)$$

A solução do problema (7) resulta um conjunto de valores para os vários α diferentes. Logo, terminamos a primeira fase deste método.

A segunda fase consiste em fazer a intersecção da pertinência da função objetivo, μ_G , com a pertinência da intersecção de todas as restrições, μ_C , da fase anterior. Assim,

$$\mu_D = \mu_G \cap \mu_C,$$

sendo $\mu_D, \mu_G, \mu_C : IR^n \rightarrow [0,1]$.

A decisão ótima no espaço de discurso é obtida com a seguinte função de pertinência

$$\mu_D(x^*) = \max_{x \in IR^n} \mu_D(x) \quad (8)$$

procurando o máximo valor dessa operação. Este método de resolução é proposto por Bellman & Zadeh (1970).

A equação (8) (Xu, 1989) pode obter um nível ótimo α^* e o ponto ótimo x^* tal que

$$\mu_G(x^*) = \max_{x \in C_{\alpha^*}} \mu_G(x) \quad (9)$$

sendo C_{α^*} o nível α^* -corte do conjunto de restrições nebulosas C .

A solução nebulosa é dada dentro de um intervalo limitado pelos valores superior e inferior fornecido pela função $\mu_G(x)$, sendo

$$\begin{aligned} m &= g_0(x^*(0)) = \min_{x \in C_0} g_0(x) \\ M &= g_0(x^*(1)) = \min_{x \in C_1} g_0(x) \end{aligned}$$

sendo que C_0 , C_1 são os níveis de corte para α igual 0 e 1, respectivamente, do conjunto de restrições nebulosas C .

Em alguns problemas de otimização nebulosa, para achar mínimos de uma função objetivo sujeito a determinadas restrições, podemos estabelecer a seguinte função:

$$\mu_G(x) = \frac{m}{g_0(x)} \quad (10)$$

Claramente, os limites superior e inferior da função objetivo são dados por

$$\begin{aligned} \mu_G^u &= 1 \\ \mu_G^l(x) &= \frac{m}{M} \end{aligned}$$

Substituindo a equação (9) na equação (10), obtemos

$$\mu_G(x^*) = \frac{1}{m} \min_{x \in C_{\alpha^*}} g_0(x) \quad (11)$$

Desta maneira, a metodologia empregada tem capacidade de otimizar um problema nebuloso pelo Método de Busca nos Limites do conjunto de soluções factíveis.

4. Relação entre os métodos

Analisando estes dois métodos foi encontrada uma relação entre os mesmos. Nesta Subseção é apresentada esta relação entre os métodos descritos nas Subseções 3.1 e 3.2. O desenvolvimento da demonstração abaixo mostra que as soluções obtidas pelos métodos acima convergem para uma única solução, mediante um intervalo definido chamado de nível de satisfação.

Lema 1: Sejam b_0, T_0 parâmetros aleatórios do primeiro método, e m, M parâmetros obtidos na resolução do segundo método. Então quando $b_0 = m$ e $T_0 = M$, os limitantes inferior e superior de $\mu_0(g_0(x))$ são os mesmos de $\mu_G(x), \forall x$.

Prova: Reescrevendo a Problema (5), podemos expressar a restrição (a) como $t_j \leq T_j(1-S)$, $\forall j = 0, 1, \dots, n$. Assim, substituindo na restrição (b), teremos $g_j(x) - T_j(1-S) \leq b_j \Rightarrow g_j(x) \leq b_j + T_j(1-S)$. Desta forma, a restrição (c) fica inoperante, pois o conjunto de restrições não depende mais de $t_j, \forall j$. Logo, a nova formulação fica

$$\begin{aligned} &\widetilde{\min} \quad S \\ \text{s.a} \quad &g_j(x) \leq b_j + T_j(1-S), \quad i = 1, 2, \dots, m, m+1, \dots, n \\ &x \geq 0 \quad \quad \quad S \geq 0 \end{aligned} \quad (12)$$

Atualizando as funções de pertinência de cada restrição, temos

$$\mu_i(x) = \begin{cases} 0, & \text{se } g_i(x) \geq b_i + T_i \\ \frac{(b_i + T_i) - g_i(x)}{T_i}, & \text{se } b_i < g_i(x) < b_i + T_i \\ 1, & \text{se } g_i(x) \leq b_i \end{cases}$$

Sendo $j = 0$, $b_0 = m$, $T_0 = M$, temos

$$\mu_i(x) = \begin{cases} 0, & \text{se } g_i(x) \geq m + M \\ \frac{(m + M) - g_0(x)}{M}, & \text{se } m < g_j(x) < m + M \\ 1, & \text{se } g_i(x) \leq m \end{cases}$$

Como $g_0(x) = M$ no caso clássico, e este é o maior valor que $g_0(x)$ pode assumir, $\forall x \geq 0$, então

$$\mu_0(g_0(x)) = \frac{(m + M) - g_0(x)}{M}, m \leq g_0(x) \leq M$$

Portanto, no máximo $\mu_0(g_0(x)) = 1$ e no mínimo $\mu_0(g_0(x)) = \frac{m}{M}$.

□

Teorema 1: Sejam as condições do Lema 1 satisfeitas, então, para ambos os métodos, no ponto ótimo $S = \mu_G \bigcap \alpha$.

Prova: (1) Se $S = 1$, temos $\min_{0 \leq j \leq n} \mu_j(g_j(x)) = 1$. Logo, $\exists x \in IR^n$ tal que $\mu_j(g_j(x)) = 1, \forall j$.

Se $\mu_D = 1$, temos $\mu_G \bigcap \mu_C = 1$. Assim, $\mu_G = 1$ e $\mu_C = 1$, para algum $x \in IR^n$. Logo, $\mu_i(g_i(x)) = 1, \forall i$, então $g_i(x) = b_i \Rightarrow \alpha = 1$.

(2) Se $S = 0$, temos $\min_{0 \leq j \leq n} \mu_j(g_j(x)) = 0$. Logo, $\exists x \in IR^n$ tal que $\mu_j(g_j(x)) = 0$, para algum j . Logo, pelo Lema 1, temos $\mu_0(g_0(x)) = 0$, se, e somente se, $m = 0$. Se $\mu_D = 0$, temos $\mu_G \bigcap \mu_C = 0$. Assim, $\mu_G = 0$ ou $\mu_C = 0$, para algum $x \in IR^n$. No primeiro caso $\mu_G = 0$, se, e somente se, $m = 0$. No segundo caso $\mu_C = 0 \Rightarrow \mu_i(g_i(x)) = 0$, para algum i , então $g_i(x) = b_i + T_i(1 - \alpha) \Rightarrow \alpha = 0$.

(3) Se $S = \lambda$, $\lambda \in (0, 1)$. Então $\min_{0 \leq j \leq n} \mu_j(g_j(x)) = \lambda$. Logo, $\exists x \in IR^n$ tal que $\mu_j(g_j(x)) = \lambda$, para algum j . Logo, pelo Lema 1, temos $\mu_0(g_0(x)) = \lambda$, se, e somente se, $\lambda \in \left[\frac{m}{M}, 1\right]$. Seja $\mu_D = \lambda$, então $\mu_G \bigcap \mu_C = \lambda$. Assim, $\mu_G = \lambda$ ou $\mu_C = \lambda$, para algum $x \in IR^n$. No primeiro caso $\mu_G = \lambda$, se, e somente se, $\lambda \in \left[\frac{m}{M}, 1\right]$. No segundo caso $\mu_C = \lambda \Rightarrow \mu_i(g_i(x)) = \lambda$, para algum i , então

$$\begin{cases} g_i(x) > b_i \Rightarrow \alpha < 1 \\ g_i(x) < b_i + T_i \Rightarrow \alpha > 0 \end{cases}$$

Portanto, $S = \mu_G \bigcap \alpha$.

□

Os resultados computacionais usando os conceitos demonstrados nessa seção estão apresentados na Seção 6. Os problemas utilizados para confirmar essa relação estão formulados na Seção 5.

5. Formulação dos exemplos numéricos

Uma formulação matemática tenta representar, através de funções, as relações existentes em um problema específico, o qual pode representar um caso real ou hipotético. As formulações apresentadas neste trabalho representam problemas reais.

5.1 Suporte de Teto com Três Barras

A Figura 8 mostra as condições de carga para um Suporte de Teto com Três Barras, o qual tem sido usado como um exemplo na literatura de otimização estrutural.

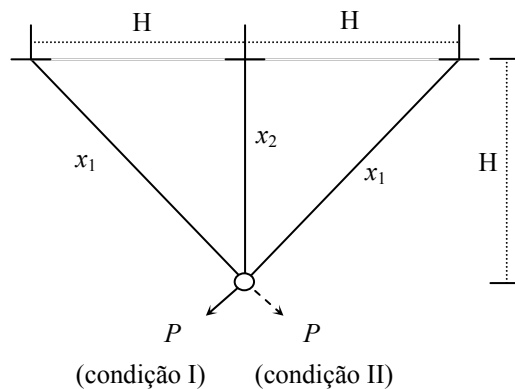


Figura 8 – Suporte de Teto com Três Barras.

Suponha que $P = 2000 \text{ kgf}$, $H = 100 \text{ cm}$, e a densidade $\gamma = 0.01 \text{ kgf/cm}^3$. Os limites e tolerâncias admissíveis de várias variáveis físicas são dadas por

$$\begin{array}{lll} \sigma^u = 2000 \text{ kgf/cm}^2 & u^u = \frac{10^5}{E} \text{ cm} & A^l = 0.1 \text{ cm}^2 \\ T_\sigma^u = 400 \text{ kgf/cm}^2 & & T_A^l = 0.02 \text{ cm}^2 \\ \sigma^l = -1500 \text{ kgf/cm}^2 & T_u^u = \frac{0.2 \times 10^5}{E} \text{ cm} & \\ T_\sigma^l = 300 \text{ kgf/cm}^2 & & \end{array}$$

O σ representa a força, u é o deslocamento vertical do nó, E representa o módulo elástico, A é a área de seção transversal, e T representa a tolerância de cada variável.

O problema de otimização nebuloso do Suporte de Teto com Três Barras pode ser formulado da seguinte forma:

$$\begin{aligned}
 \min \quad & W = 2\sqrt{2}x_1 + x_2 \\
 \text{s.a.} \quad & \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} \lesssim 1 \\
 & \frac{\sqrt{2}x_1}{\sqrt{2x_1^2 + 2x_1x_2}} \lesssim 1 \\
 & \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} \lesssim \frac{3}{4} \\
 & \frac{1}{x_1 + \sqrt{2}x_2} \lesssim \frac{1}{2} \\
 & x_1, x_2 \gtrsim \frac{1}{10}
 \end{aligned} \tag{13}$$

5.2 Anteparo Ondulado

A Figura 9 mostra um Anteparo Transversal Verticalmente Ondulado para um tanque de 65.000 dwt, o qual está construído de acordo com as regras das especificações relevantes ao projeto de navios.

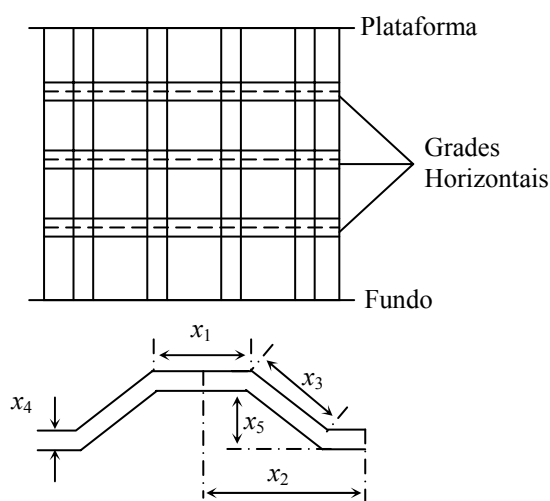


Figura 9 – Anteparo Ondulado.

Dados do projeto são: largura do anteparo $B = 21.12$ m, altura do painel inferior $H = 4.5$ m, e densidade 7.85×10^{-3} t/mm-m². As posições das colunas horizontais e a largura do casco são fixadas, de modo que a forma da corruação dependa do painel inferior.

As variáveis do projeto são: x_1 = largura do aro (mm); x_2 = semi-largura da corruação (mm); x_3 = comprimento da armação (mm); x_4 = espessura da placa no painel inferior (mm); e x_5 = profundidade da corruação (mm).

A função objetivo examina o peso do painel inferior e está sujeita a 6 (seis) restrições nebulosas no módulo da secção, o momento de inércia, a espessura mínima da chapa do painel inferior e uma restrição clássica que relaciona as dimensões da seção de corrugação.

Logo, o problema de otimização nebulosa do casco ondulado pode ser formulado como segue:

$$\begin{aligned}
 \min \quad & W = 0.746064(x_1 + x_3)x_4x_2^{-1} \\
 \text{s.a.} \quad & x_5x_4(3x_1 + x_3)x_2^{-1} \gtrsim 18871.4 \\
 & x_5x_4(3x_1 + x_3)(x_5 + 2x_4)x_2^{-1} \gtrsim 540408 \\
 & x_4 - 0.017297x_1 \gtrsim 2.5 \\
 & x_4 - 0.017297x_3 \gtrsim 2.5 \\
 & x_4 - 5.5 \gtrsim 6.5 \\
 & \arcsen(x_5x_3^{-1}) \gtrsim 40^\circ \\
 & x_5^2 \leq x_3^2 - (x_2 - x_1)^2 \\
 & x_1, x_2, x_3, x_4, x_5 > 0
 \end{aligned} \tag{14}$$

6. Resultados computacionais

Nesta Seção estão os resultados computacionais dos exemplos numéricos da seção anterior. Os resultados aparecem na mesma ordem que seus respectivos exemplos numéricos. Todos os modelos clássicos foram resolvidos usando a função **fmincon** do *ToolBox Optimization* do programa **MatLab 6.1**, o qual soluciona problemas de programação matemática restrita, e 100% é o nível de satisfação destas soluções.

Tabela 1 – Resultado do Exemplo Numérico 3.1 dos modelos para comparação.

Modelo	x_1	x_2	Função Objetivo	Iterações		
				RTV	RNV	Solução
Modelo Clássico	0.667	0.943	2.828	-	-	5
Modelo <i>Fuzzy</i> de Xu	0.649	0.917	2.752	6	5	12
Modelo <i>Fuzzy</i> do Trappey, Liu e Chang	0.649	0.917	2.752	6	5	7

O Modelo Clássico convergiu para uma solução ótima, apresentada na Tabela 2, em 5 iterações. Conforme Silva & Yamakami (2004), para iniciar os modelos *fuzzy* precisa-se obter a solução ótima para dois casos: (1) restrições totalmente violadas (RTV); e (2) restrições não violadas (RNV). O primeiro caso convergiu para uma solução ótima em 6 iterações, onde o ponto encontrado foi [0.556 0.786] com valor de Função Objetivo igual a 2.357, enquanto o segundo caso convergiu em 5 iterações, onde este caso tem a mesma solução ótima do Modelo Clássico. Assim, os métodos *fuzzy* são inicializados, para este problema, com 11 iterações. O Modelo *Fuzzy* de Xu convergiu para uma solução ótima, apresentada na Tabela 2, em 23 iterações, onde o nível de satisfação desta solução é de 85.64%. O Modelo *Fuzzy* do Trappey, Liu e Chang obtiveram uma satisfação de 86.05%, e convergiu para uma solução ótima, descrita na Tabela 2, em 18 iterações.

Tabela 2 – Resultado do Exemplo Numérico 3.2 dos modelos para comparação.

Modelo	x_1	x_2	x_3	x_4	x_5	Função Objetivo	It.
Modelo Clássico	643.129	997.166	643.129	13.624	536.911	13.111	16
Modelo <i>Fuzzy</i> de Xu	637.050	987.439	637.050	13.469	532.034	12.966	71
Modelo <i>Fuzzy</i> de Trappey, Liu e Chang	638.305	991.212	638.305	13.489	531.874	12.961	40

Na Tabela 3 o Modelo Clássico convergiu para uma solução ótima em 16 iterações. Na inicialização dos modelos *fuzzy* o primeiro caso convergiu para uma solução ótima em 15 iterações, onde o ponto ótimo foi [579.097 894.931 579.097 12.017 485.388] com valor de Função Objetivo igual a 11.602, enquanto o segundo caso convergiu em 16 iterações. Assim, os métodos *fuzzy* são inicializados, para este problema, com 31 iterações. O Modelo *Fuzzy* de Xu convergiu para uma solução ótima apresentada na Tabela 3 em 71 iterações, onde o nível de satisfação desta solução é de 89.49%. O Modelo *Fuzzy* do Trappey, Liu e Chang obtiveram uma satisfação de 89.64%, e convergiu para uma solução ótima, descrita na Tabela 3, em 40 iterações.

7. Conclusões

Neste trabalho foi demonstrada uma relação existente entre dois métodos de programação não-linear *fuzzy*. Esta relação mostra que estes métodos tendem a convergir para o mesmo ponto ótimo, mediante as condições de que o nível de aspiração e tolerância máxima do método *fuzzy* apresentado na Subseção 3.1 sejam iguais às soluções do Problema (7) para $\alpha = 0$, que representa o problema totalmente violado, e para $\alpha = 1$, que representa o problema não violado, respectivamente. Sendo que esse Problema (7) faz parte da primeira fase do método descrito na Subseção 3.2 para resolver modelos não-lineares *fuzzy*. Uma outra análise que se pode retirar desta formulação está associada à formulação do problema a ser otimizado, pois em problemas com formulação convexa as soluções de ambos tendem a convergir para valores dentro de uma determinada vizinhança.

Diante dos resultados expostos na seção anterior, pode-se inferir que as soluções dos Modelos *Fuzzy* fornecem resultados melhores que as soluções dos modelos clássicos. O nível de satisfação dos Modelos *Fuzzy* não atingem 100%, mas em todos os resultados acima obtiveram satisfação superior a 80%. Todos os exemplos numéricos apresentados neste artigo convergiram para uma solução ótima. Os Modelos Clássicos convergiram com uma quantidade menor de iterações, porém resolve um problema formulado de forma exata. O número maior de iterações para os Modelos *Fuzzy* convergirem para uma solução ótima é compensado pelo resultado obtido e pela menor complicação de introduzir dados incertos, referentes a problemas reais, sem aumentar demasiadamente o esforço computacional referentes a problemas reais.

A previsão de trabalhos futuros está associada na utilização desses métodos de programação não-linear *fuzzy* em problemas reais. A principal intenção é observar o comportamento desses métodos e analisar a convergência de ambos. Outro ponto importante é desenvolver outros métodos *fuzzy* que resolvam estes mesmos problemas sem o auxílio de um ponto inicial gerado por métodos clássicos que resolver problemas sem incerteza.

Agradecimentos

Os autores agradecem ao CAPES pelo suporte financeiro.

Referências Bibliográficas

- (1) Bazaraa, M.S.; Sherali, H.D. & Shetty, C.M. (1993). *Nonlinear Programming – Theory and Algorithms*. Segunda edição, John Wiley & Sons, New York.
- (2) Bellman, R.E. & Zadeh, L.A. (1970). Decision-making in a fuzzy environment. *Management Science*, **17**(4), B141-B164.
- (3) Ekel, P.Y. (2002). Fuzzy Sets and Models of Decision Making. *Computers and Mathematics with Applications*, **44**, 863-875.
- (4) Galperin, E.A. & Ekel, P.Y. (2005). Synthetic Realization Approach to Fuzzy Global Optimization via Gamma Algorithm. *Mathematical and Computer Modelling*, **41**, 1457-1468.
- (5) Ferreira, P.A.V. (2002). *Otimização Não-linear*. Disponível em <<http://www.dt.fee.unicamp.br:80/~valente/ia543.html>>.
- (6) Klir, G.J. & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic – Theory and Applications*. Prentice-Hall, New Jersey.
- (7) Lai, Y.J. & Hwang, C.L. (1992). *Fuzzy Mathematical Programming: Methods and Applications*. Vol. 394 de Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin.
- (8) Lee, Y.H.; Yang, B.H. & Moon, K.S. (1999). An economic machining process model using fuzzy non-linear programming and neural network. *International Journal Production Research*, **37**(4), 835-847.
- (9) Luenberger, D.G. (1984). *Linear and Nonlinear Programming*. Segunda edição, Addison-Wesley, Massachusetts.
- (10) Pedrycz, W. & Gomide, F. (1998). *An introduction and fuzzy sets: Analysis and Design*. A Bradford Book.
- (11) Silva, R.C. (2005). Contribuições ao estudo de programação não-linear com incertezas. Dissertação de mestrado, FEEC – UNICAMP, Campinas, Maio.
- (12) Silva, R.C. & Yamakami, A. (2004). Uma contribuição para programação não-linear fuzzy. Aceito no CBA 2004, em Gramado – RS, 21 a 27 de Setembro de 2004.
- (13) Silva, R.C. & Cantão, L.A.P. (2004). Comparação entre modelos clássicos e fuzzy para problemas de programação não-linear. Aceito no SBPO 2004, em São João Del Rei – MG, 21 a 27 de Novembro de 2004.
- (14) Trappey, J.-F.C.; Liu, C.R. & Chang, T.-C. (1988). Fuzzy non-linear programming: Theory and application in manufacturing. *International Journal Production Research*, **26**(5), 975-985.
- (15) Xu, C. (1989). Fuzzy optimization of structures by the two-phases method. *Computers & Structures*, **31**(4), 575-580.
- (16) Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, **8**, 338-353.